

UNIVERSITY OF
ILLINOIS LIBRARY
AT URBANA-CHAMPAIGN

MATHEMATICS



Digitized by the Internet Archive
in 2013

<http://archive.org/details/performanceevalu891yama>

PERFORMANCE EVALUATION OF MULTIPROCESSOR SYSTEMS
CONTAINING SPECIAL PURPOSE PROCESSORS

by
Haruaki Yamazaki

January 1978



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

The Library of the
MAR 27 1978
University of Illinois
at Urbana-Champaign

PERFORMANCE EVALUATION OF MULTIPROCESSOR SYSTEMS

by

Haruaki Yamazaki

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

January 1978

*

This work was supported in part by the National Science Foundation under grant MCS 73-07980 A03.

TABLE OF CONTENTS

Chapter		Page
1	INTRODUCTION.....	1
2	NOTATIONS.....	3
3	THE SYSTEMS WITH INDEPENDENT INPUT PROCESS.....	5
	3.1 The Deterministic Task Execution Time.....	5
	3.1.1 The Performance of a Multiprocessor System with General Purpose Processors.....	7
	3.1.2 The Performance of a Multiprocessor System with Special Purpose Processors.....	14
	3.2 The Heavy Traffic Approximation with Exponential Execution Time.....	18
	3.2.1 The Performance of a Multiprocessor System with General Purpose Processors.....	18
	3.2.2 The Performance of a Multiprocessor System with Special Purpose Processors.....	21
4	THE SYSTEM WITH SERIAL INPUT PROCESS.....	28
	4.1 The Performance of a Multiprocessor System with General Purpose Processors.....	28
	4.2 The Performance of a Multiprocessor System with Special Purpose Processors.....	30
	LIST OF REFERENCES.....	34
	APPENDIX.....	35

LIST OF FIGURES

Figure		Page
3.1	Decomposition of a Job.....	6
3.2	System with General Purpose Processors.....	8
3.3	Time Slot.....	9
3.4	Curves \bar{W}_S and \bar{W}_g	15
3.5	Multiprocessor System with Special Purpose Processors...	16
3.6	Equivalent Single Processor Model.....	20
3.7	\bar{W}_g and \bar{W}_S under Heavily Loaded Condition.....	26
3.8	\bar{W}_g and \bar{W}_S under Heavily Loaded Condition.....	27
4.1	System with General Purpose Processors.....	29
4.2	Multiprocessor System with Special Purpose Processors...	31
4.3	\bar{W}_g and \bar{W}_S under Heavily Loaded Condition.....	33

1. INTRODUCTION

In recent years, large and complex multiprocessor systems have been designed and implemented. The architectural differences among these systems can be characterized in terms of the way to interconnect functional units [1]. Some of them are common bus systems, or crossbar switch systems (e.g., Cmmp [2]), or multiport-memory systems (e.g., Prime [3]). We focus our attention here on the types of functional units (processors) in the system. Some multiprocessor systems consist of identical general purpose processors, which share the input job load under the control of the job scheduler. Other multiprocessor systems consist of special purpose processors (or functionally dedicated processors) each of which is designed to execute a particular type of job or task. (For example, these processors may be file management processors, input-output control processors, and processors for high speed computations, etc.) These special purpose processors may be implemented in hardware, firmware or software.

In this paper, both types of multiprocessor systems are modelled queueing theoretically and their performance is evaluated. Our purpose here is not to extend known results in the queueing theory itself, but to determine the architectural merits of the two types of systems. Moreover, in the case of the multiprocessor system with special purpose processors, the optimal architecture are discussed.

The notations which are necessary in our discussions are defined in section 2. In section 3, we discuss the models where the arrivals of

different types of tasks are statistically independent. In section 4, queueing networks are used to model systems in which the execution of some tasks cannot be started before the completion of some other tasks.

2. NOTATIONS

Let us assume that jobs arriving for service of the multi-processors system may be decomposed into a number of different tasks. There are altogether $N+1$ different types of tasks. (For example, consider a system in which some jobs may be decomposed into input tasks followed by compilation, computation and output tasks while the other jobs may be decomposed into input, sorting, merging and output tasks. In this case, we say that there are 6 different types of tasks.) We refer to the set of type i tasks as task set \mathcal{T}_i .

It suffices here to consider the relative speeds of the processors. In particular, to compare the merits of a system consisting of different special purpose processors with the system consisting of identical general purpose processors, we measure their relative speeds. The relative speed of a special purpose processor with respect to a general purpose processor is called the capacity of the special purpose processor. For example, if a task takes $1/\mu$ units of time to be completed by a general purpose processor, then it takes $(1/\mu)(1/C)$ units of time to be completed by a special purpose processor with capacity C .

We refer the time required to complete a task in a given system as the execution time (or service time) of the task on that system. Particularly, the execution time of the task on a general purpose processor is called the amount of work for that task. Hence, if a special purpose processor with capacity C completes the given task within t sec, then the amount of work for that task is $t \cdot C$ units.

We measure the effectiveness of a multiprocessor system by the total amount of work remaining in the system. That is, the total time required for a general purpose processor to complete all tasks being served and waiting for service in the system. This performance measure is chosen since it is dependent only on the architecture of the system but not on the queueing discipline used to schedule tasks on processors.

Let \bar{n}_q be the mean queue length in a system consisting of a processor with capacity C , and $E(S_r)$ be the mean residual time (that is the remaining execution time of the task in progress). The average total amount of work remaining in the system is given by

$$\frac{1}{\mu} \bar{n}_q + CE(S_r) \quad (2.1)$$

where $1/\mu$ is the average amount of work for the task. (Note that \bar{n}_q and $E(S_r)$ also depend on C .)

3. THE SYSTEMS WITH INDEPENDENT INPUT PROCESS

In this section, we consider the queueing models of multiprocessor systems in which the arrival processes of the different types of tasks are statistically independent. It is difficult to analyze the general behavior of multiprocessor queues except for the simple case where the arrival process is Poisson and service time is exponentially distributed. The assumption of exponential distribution for service time being invalid in our case, we are forced to use various approximation methods. Again, our purpose in this section is to evaluate the relative effectiveness of different multiprocessor system architectures.

3.1 The Deterministic Task Execution Time

In this model, we assume that the amount of work required for all tasks is constant and identical. Furthermore, the interarrival time of jobs is exponentially distributed. Each of these jobs is decomposed into a number of tasks as shown in Figure 3.1. We say that these tasks are generated by the job. Let T_i denote the number of tasks of type i generated by a job. Then the total number of tasks generated by the job is $T = T_0 + T_1 + \dots + T_N$. We assume, furthermore, that T_i 's are statistically independent random variables. Let us denote the generating function of random variable T_i by $A_i(Z)$. Then, the generating function

of T is $\prod_{i=0}^N A_i(Z)$.

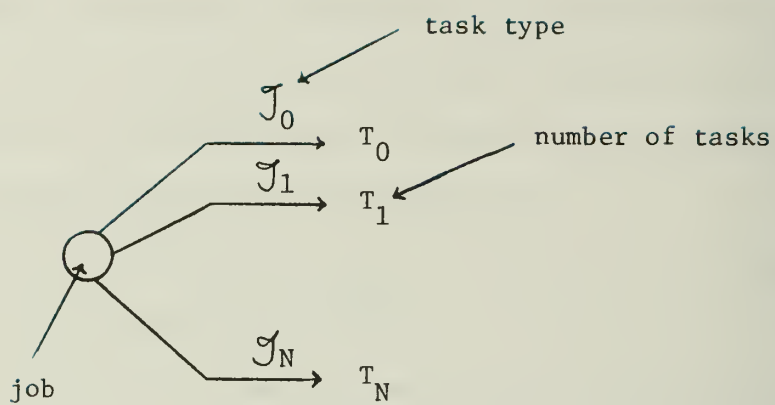


Figure 3.1 Decomposition of a Job

3.1.1 The Performance of a Multiprocessor System with General Purpose Processors

This system consists of m general purpose processors as described in Figure 3.2. Since all processors are identical, a task can be executed by any of the m processors. Therefore, the job scheduler assigns a waiting task to any processor when it becomes idle. When all processors are busy, the task joins the common queue.

Let us define the amount of work for a task as a unit of time or a time slot (see Figure 3.3). If the interarrival time of jobs is sufficiently long compared with this time slot, then we can approximate the exponential distribution of the interarrival time by a geometric distribution. Specifically, the probability that there are n jobs arrived within the time duration Δt starting from time t is:

$$p_n = \Pr \{n \text{ jobs arrived in } [t, t+\Delta t]\} = \frac{(\lambda \Delta t)^n}{n!} e^{-\lambda(\Delta t)}.$$

If we assume that λ is sufficiently small ($\lambda \ll 1$), then we can approximate this probability distribution as:

$$\begin{cases} p_0 = 1 - \lambda \Delta t \\ p_1 = \lambda \Delta t \\ p_n = 0 \quad \text{for } n = 2, 3, \dots \end{cases}$$

Let $\Delta t = 1$. The expression above becomes:

$$\begin{cases} p_0 = 1 - \lambda \\ p_1 = \lambda \end{cases}$$

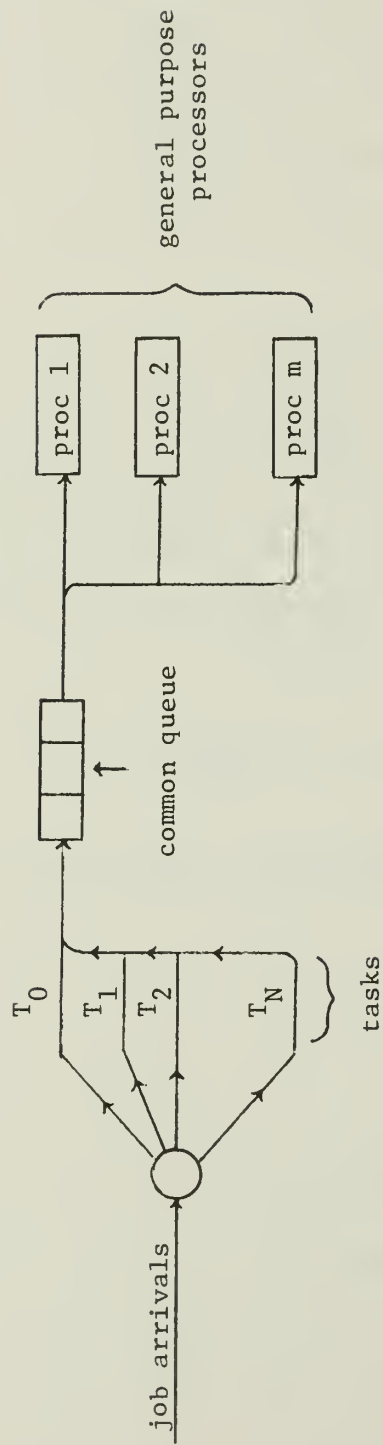


Figure 3.2 System with General Purpose Processors

a time slot

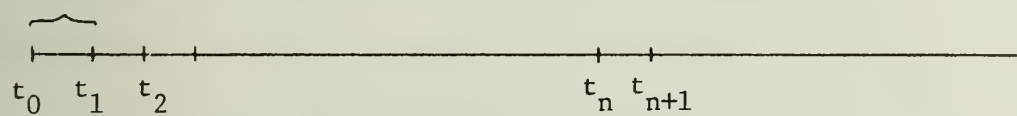


Figure 3.3 Time Slot

Let the generating function of the total number of tasks arrived within a time slot be denoted by $A(Z)$. $A(Z)$ is given by:

$$A(Z) = (1-\lambda) + \lambda \prod_{i=0}^N A_i(Z) \quad (3.1)$$

We assume that the job scheduler assigns tasks to the processors at the beginning of each time slot. If t_n is the n th epoch of this time slot, the total number of tasks at $t_n + 0^+$ forms a markov chain. Since the execution time of a task is 1 time slot, the total number of tasks at $t_n + 0^+$ is equivalent to the amount of work in the system at $t_n + 0^+$. Let the probability of the arrival of j tasks during one time slot be P_j (i.e., $A(Z) = \sum_{j=0}^{\infty} P_j Z^j$). Assuming that the number of processor m is larger than $\lambda E(T)$, we solve the mean number of the tasks in the system at the equilibrium.

Let π_{ij} be the transition probability of the total number of tasks in the system, then

$$\pi_{ij} = \begin{cases} 0 & \text{for } j < i - m \\ P_j & \text{for } i < m \\ P_{j-i+m} & \text{for } i \geq m, j \geq i - m \end{cases} \quad (3.2)$$

Since at the equilibrium, the probability of having j tasks in the system is

$$\pi_j = \sum_{i=0}^{\infty} \pi_{ij} \pi_i,$$

the generating function $P(Z)$ is

$$P(Z) = \sum_{j=0}^{\infty} \pi_j Z^j = \sum_{j=0}^{\infty} Z^j \sum_{i=0}^{\infty} \pi_{ij} \pi_i = \sum_{i=0}^{\infty} \pi_i \sum_{j=0}^{\infty} \pi_{ij} Z^j.$$

From the condition in (3.2), we get:

$$P(Z) = \prod_0 \sum_{j=0}^{\infty} \prod_{0j} Z^j + \prod_1 \sum_{j=0}^{\infty} \prod_{1j} Z^j + \dots + \prod_{m-1} \sum_{j=0}^{\infty} \prod_{m-1j} Z^j + \sum_{i=m}^{\infty} \prod_i \sum_{j=0}^{\infty} \prod_{ij} Z^j.$$

Thus,

$$P(Z) = \sum_{i=0}^{m-1} \prod_i \sum_{j=0}^{\infty} P_j Z^j + \sum_{i=m}^{\infty} \prod_i \sum_{j=i-m}^{\infty} P_{j-i+m} \cdot Z^{j-i+m} \cdot Z^{i-m}.$$

$$\text{Since } A(Z) = \sum_{i=0}^{\infty} P_i Z^i,$$

$$\begin{aligned} P(Z) &= \sum_{i=0}^{m-1} \prod_i A(Z) + \sum_{i=m}^{\infty} \prod_i A(Z) Z^{i-m} \\ &= \sum_{i=0}^{m-1} \prod_i A(Z) + \frac{A(Z)}{Z^m} (P(Z) - \sum_{i=0}^{m-1} \prod_i Z^i). \end{aligned}$$

In other words,

$$P(Z) = \frac{A(Z)}{A(Z) - Z^m} \left(\sum_{i=0}^{m-1} \prod_i Z^i - Z^m \sum_{i=0}^{m-1} \prod_i \right).$$

$$\text{Let } F(Z) = \sum_{i=0}^{m-1} \prod_i (Z^i - Z^m), \text{ then}$$

$$P(Z) = \frac{A(Z)F(Z)}{A(Z) - Z^m}. \quad (3.3)$$

$$\text{Since } \lim_{Z \rightarrow 1} P(Z) = 1 \text{ and } \lim_{Z \rightarrow 1} \frac{A(Z)F(Z)}{A(Z) - Z^m} = 1,$$

$$\lim_{Z \rightarrow 1} \frac{A'(Z)F(Z) + A(Z)F'(Z)}{A'(Z) - mZ^{m-1}} = \frac{A'(1)F(1) + A(1)F'(1)}{A'(1) - m} = 1.$$

In other words,

$$A'(1)F(1) + F'(1) = A'(1) - m,$$

But $F(1) = 0$. Hence

$$F'(1) = A'(1) - m. \quad (3.4)$$

However,

$$F'(Z) = \sum_{i=0}^{m-1} (iZ^{i-1} - mZ^{m-1}) \Pi_i,$$

and

$$-F'(1) = \sum_{i=0}^{m-1} (m-i) \Pi_i.$$

The sum in the right hand side of the equation above is the mean number of idle processors. Then, $m+F'(1)$ is the mean number of busy processors. Therefore, according to Eq. (3.4)

$$A'(1) = E\{\text{number of busy processors}\}.$$

The average number of tasks in the system at each epoch is derived in appendix. It is given by:

$$P'(1) = \bar{n} = \sum_{i=0}^{m-2} \frac{1}{(1-Z_i)} - \frac{m(m-1)}{2(m-A'(1))} + A'(1) + \frac{A''(1)}{2(m-A'(1))}, \quad (3.5)$$

where Z_0, Z_1, \dots, Z_{m-2} are the zeros of $(1 - \frac{Z^m}{A(Z)})$ within the unit circle. If we assume that every job is decomposed into a number of tasks that are integer numbers of m , then we can solve explicitly for \bar{n} . In other words, we assume that $\prod_{i=1}^m A_i(Z)$ is a polynomial of Z^m . (This model can be reduced to the single processor case. But here we will derive the explicit expression using eq. (3.5).) In this case, $\Pi_i = 0$ for $i = 1, 2, \dots, m-1$, and by appendix the first term in eq. (3.5) can be expressed as $\frac{F''(1)}{2F'(1)}$.

Since

$$F''(1) = \frac{d^2 F(Z)}{dZ} = \sum_{i=1}^{m-1} \Pi_i \{i(i-1)Z^{i-2} - m(m-1)Z^{m-2}\} \Big|_{Z=1},$$

$$F''(1) + m(m-1) = \sum_{i=1}^{m-1} i(i-1) \Pi_i + \sum_{i=m}^{\infty} m(m-1) \Pi_i .$$

Let $\rho = \sum_{i=m}^{\infty} \Pi_i$. Then

$$F''(1) + m(m-1) = \sum_{i=1}^{m-1} i(i-1) \Pi_i + m(m-1)\rho .$$

Therefore

$$F''(1) = m(m-1)(\rho-1) .$$

Meanwhile

$$A'(1) = \sum_{i=1}^{m-1} i \Pi_i + \sum_{i=m}^{\infty} m \Pi_i = m\rho .$$

Therefore

$$\frac{F''(1)}{2F'(1)} = \frac{m(m-1)(\rho-1)}{2m(\rho-1)} = \frac{m-1}{2} .$$

Thus we get

$$\bar{n} = \frac{m-1}{2} - \frac{m(m-1)}{2(m-A'(1))} + A'(1) + \frac{A''(1)}{2(m-A'(1))} . \quad (3.6)$$

According to eq. (3.4), the mean number of busy processors is given by $A'(1)$. Hence, $\bar{n} - A'(1)$ tasks are in the queue. For an arbitrary time t , the mean residual execution time is given as $1/2$. The average amount of work in the system is given by:

$$\bar{W}_g = \bar{n} - A'(1) + \frac{1}{2} A'(1)$$

or

$$\bar{W}_g = \frac{m-1}{2} - \frac{m(m-1)}{2(m-A'(1))} + \frac{A'(1)}{2} + \frac{A''(1)}{2(m-A'(1))} \quad (3.7)$$

where $m > A'(1)$. Figure 3.4 describes the behavior of \bar{W}_g when $A(Z) = 1 - \lambda + \lambda Z^6$, and $m = N + 1 = 3$.

3.1.2 The Performance of a Multiprocessor System with Special Purpose Processors

This system consists of $N + 1$ subsystem of special purpose processors as described in Figure 3.5, where $m_0 + m_1 + \dots + m_N = m$. Since a special purpose processor can execute only one type of task, the job scheduler assigns a type i task to i th subsystem or let it join the i th queue.

Let us consider the number of tasks in the i th subsystem. If we let C_i be the capacity of each processor in this subsystem ($C_i \geq 1$), then the execution time of a task is $1/C_i$. Therefore, if we define this execution time as a time slot, then the number of tasks arrived to this processor during one time slot is $(\lambda/C_i) T_i$.

Let $B_i(Z)$ be the generating function of the number of tasks arrived during the execution of a task, and $P_i(Z)$ be a generating function of the number of tasks in i th subsystem. Then the same discussion as in the previous section can be applied. According to eq. (3.3),

$$P_i(Z) = \frac{B_i(Z)F_i(Z)}{B_i(Z) - Z^{\frac{m_i}{C_i}}}$$

where

$$B_i(Z) = 1 - \frac{\lambda}{C_i} + \frac{\lambda}{C_i} A_i(Z) .$$

From eq. (3.5), the average number of tasks in i th subsystem \bar{n}_i is given by:

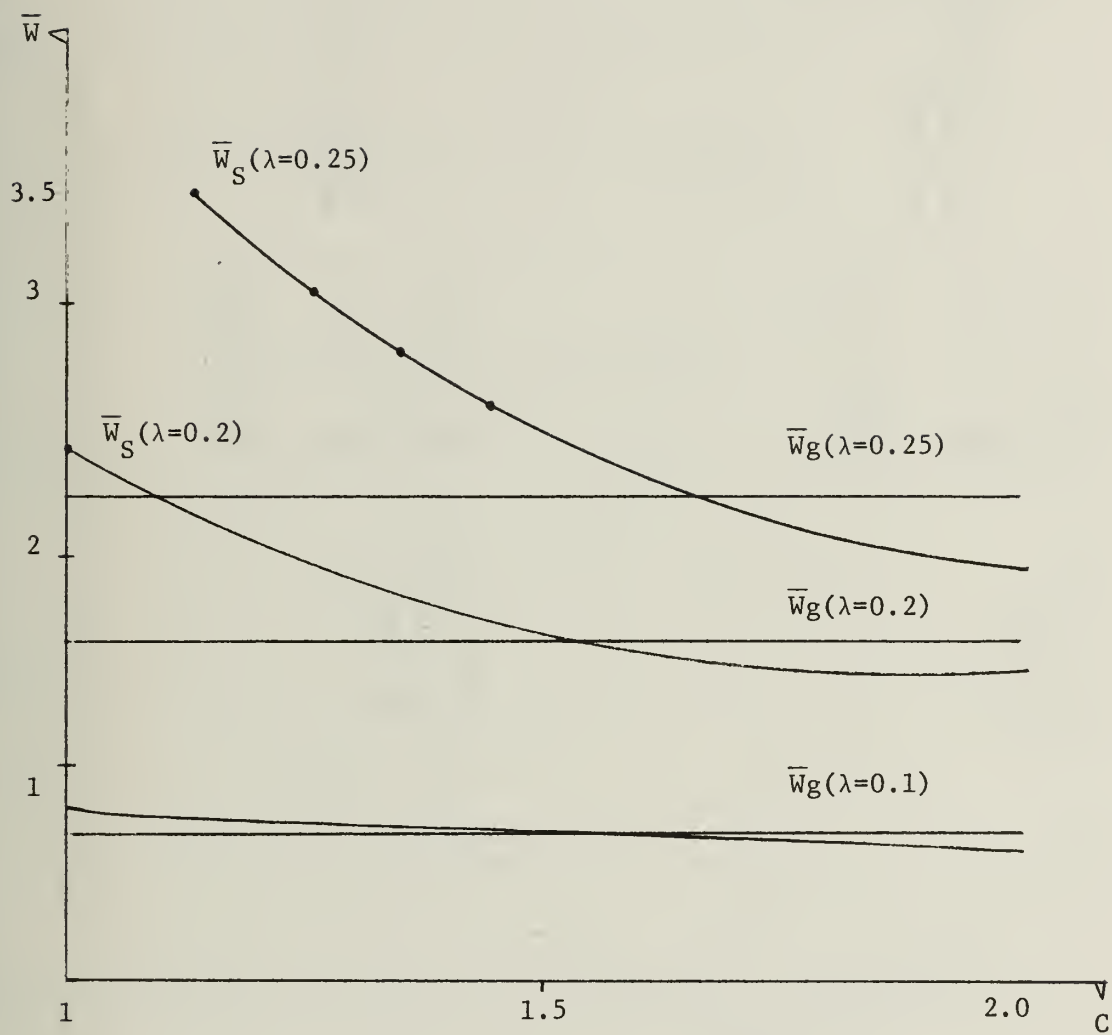


Figure 3.4 Curves \bar{W}_S and \bar{W}_g

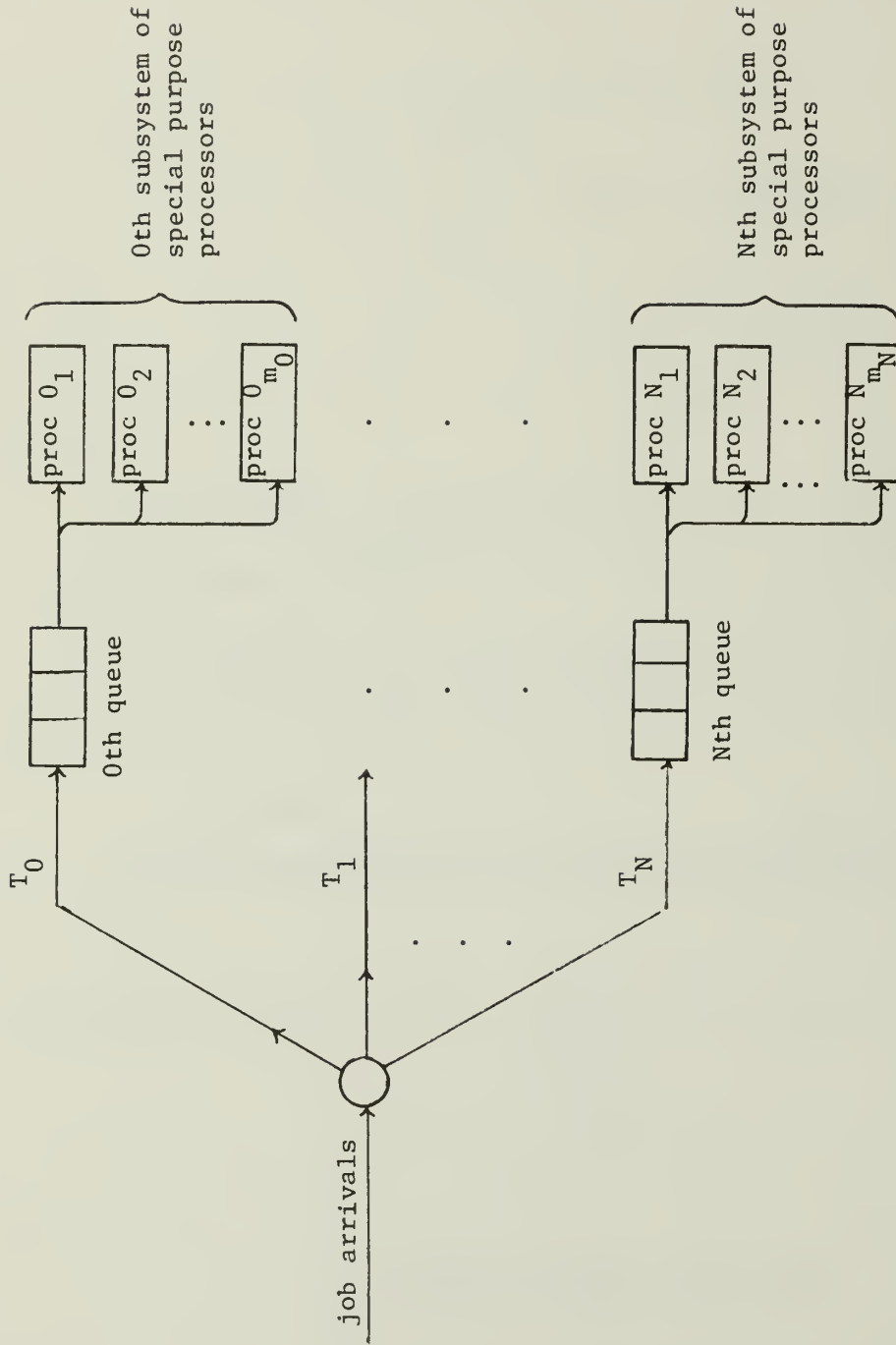


Figure 3.5 Multiprocessor System with Special Purpose Processors

$$P'_i(1) = \bar{n}_i = \sum_{j=0}^{m_i-2} \frac{1}{1-Z_j} - \frac{m_i(m_i-1)}{2(m_i-B'_i(1))} + \frac{B''_i(1)}{2(m_i-B'_i(1))} .$$

Thus, if we assume $A_i(Z)$ is a polynomial of Z^i , the average amount of work in the i th subsystem \bar{W}_i is given by equation (3.7).

$$\bar{W}_i = \frac{m_i-1}{2} - \frac{m_i(m_i-1)}{2(m_i-B'_i(1))} + \frac{B'_i(1)}{2} + \frac{B''_i(1)}{2(m_i-B'_i(1))} .$$

where

$$B_i(Z) = 1 - \lambda/C_i + (\lambda/C_i) A_i(Z) .$$

Moreover, the total mean amount of work in this system \bar{W}_s is given by:

$$\bar{W}_s = \sum_{i=0}^N \left\{ \frac{m_i-1}{2} - \frac{m_i(m_i-1)}{2(m_i-B'_i(1))} + \frac{B'_i(1)}{2} + \frac{B''_i(1)}{2(m_i-B'_i(1))} \right\} \quad (3.8)$$

where

$$m = \sum_{i=0}^N m_i$$

In particular, if $C_i = C$, $m_i = 1$, $N+1 = m$, then eq. (3.8) becomes

$$\bar{W}_s = \frac{\lambda}{2} \sum_{i=1}^m \left\{ \frac{A'_i(1)}{C} + \frac{CA''_i(1)}{(C-\lambda \cdot A'_i(1))} \right\}$$

Figure 3.4 also shows the behavior of the curves \bar{W}_s in the case $N+1 = m = 3$, $A_i(Z) = Z^i$, $m_i = 1$ for $i = 1, 2, 3$. That is,

$$\bar{W}_g = 3\lambda - 1 + \frac{\lambda+1}{1-2\lambda}$$

$$\bar{W}_s = \frac{\lambda}{2} \left\{ \frac{6}{C} + \frac{2C}{C-2\lambda} + \frac{6C}{C-3\lambda} \right\}$$

where $\lambda \leq \frac{1}{2}$ and $\lambda \leq C/3$.

3.2 The Heavy Traffic Approximation with Exponential Execution Time

In this model, we assume the amount of work required for type i tasks is exponentially distributed with parameter α_i . Furthermore, we assume the arrived job is decomposed into $N + 1$ tasks of different types. In other words, if we let S_i be the amount of work required for type i job, then the total amount of work generated by the job is $S = S_0 + S_1 + \dots + S_N$ where S_i 's are assumed to be statistically independent random variables. Moreover, we assume the interarrival time of two jobs is exponentially distributed with parameter λ , and the system is heavily loaded.

3.2.1 The Performance of a Multiprocessor System with General Purpose Processors

This system is similar to the one described in Figure 3.2. Let us denote by t_n the epoch when n th job arrived. Let W_n be the amount of work remaining at t_n , A_n be the amount of work completed and B_n be the amount of work arrived in the duration $[t_{n-1}, t_n]$. Then W_{n+1} can be expressed as

$$W_{n+1} = \begin{cases} W_n + B_n - A_n & \text{if } W_n + B_n - A_n \geq 0 \\ 0 & \text{if } W_n + B_n - A_n \leq 0 \end{cases} \quad (3.9)$$

Here, we assume heavy traffic condition. That is: the number of tasks arrived during $[t_{n-1}, t_n]$ is sufficiently large so that all m processors are busy all the time. Therefore, A_n is independent of n .

Moreover, A_n has the same distribution as mA where A is the interarrival time of two jobs. Since A is exponentially distributed with parameter λ , mA is also exponentially distributed with parameter λ/m . Also, B_n has the same distribution as S , we can write eq. (3.9) as:

$$W_{n+1} = \begin{cases} W_n + (S - mA) & \text{if } W_n + S - mA \geq 0 \\ 0 & \text{if } W_n + S - mA < 0 \end{cases} \quad (3.10)$$

However, the expression in the right hand side of this equation is the waiting time of the $(n+1)$ th job in the case when the system consists of one processor, the execution time of a job is S , and the interarrival time of jobs is mA . Thus, we see that the mean waiting time in the single server system shown in Figure 3.6 is equivalent to the average amount of work remaining in our original model.

Since $\Pr\{A \leq t\} = 1 - e^{-\lambda t}$, and $S = S_0 + S_1 + \dots + S_n$ where $\Pr\{S_i \leq x\} = 1 - e^{-\alpha_i x}$,

$$E[mA] = \frac{m}{\lambda}$$

$$E[S] = \sum_{i=0}^N \frac{1}{\alpha_i}$$

$$E[S^2] = \left(\sum_{i=0}^N \frac{1}{\alpha_i} \right)^2 + \sum_{i=0}^N \left(\frac{1}{\alpha_i} \right)^2.$$

In the case of single server, the mean queue length and the mean residual time are given by [4]:

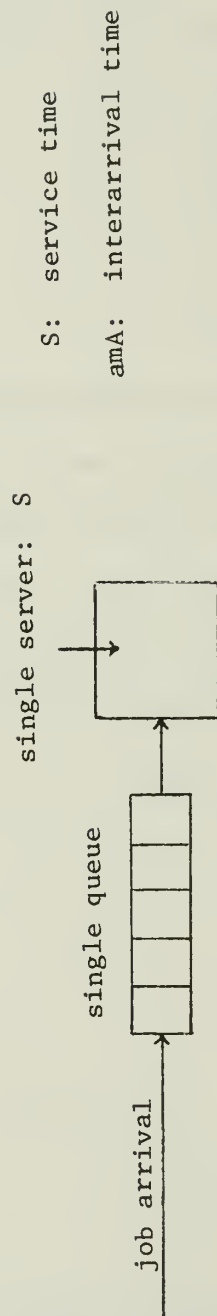


Figure 3.6 Equivalent Single Processor Model

$$\bar{n}_g = \frac{(\lambda/m)^2 E(S^2)}{2(1-\rho)}$$

$$E(S_r) = \frac{(\lambda/m)E(S^2)}{2}$$

where $\rho = (\lambda/m)E(S) = (\lambda/m) \sum_{i=0}^N (\frac{1}{\alpha_i})$. Therefore, as mentioned in

Section 2, the mean waiting time in queue of this single server system is

$$E(S) \bar{n}_g + E(S_r) = \frac{(\lambda/m)E(S^2)}{2} (\frac{\rho}{1-\rho} + 1)$$

Moreover, the mean amount of work remaining in our original system \bar{W}_g is:

$$\begin{aligned} \bar{W}_g &= \frac{(\lambda/m)E(S^2)}{2} (\frac{1}{1-\rho}) \\ &= \frac{(\lambda/m)}{2} \left(\left(\sum_{i=0}^N \frac{1}{\alpha_i} \right)^2 + \sum_{i=0}^N \left(\frac{1}{\alpha_i} \right)^2 \right) (\frac{1}{1-\rho}) \end{aligned}$$

where

$$\rho = (\lambda/m) \sum_{i=0}^N (\frac{1}{\alpha_i})$$

or

$$\bar{W}_g = \frac{1}{2(1-\rho)} \left\{ \rho \sum_{i=0}^N \left(\frac{1}{\alpha_i} \right) + (\lambda/m) \sum_{i=0}^N \left(\frac{1}{\alpha_i} \right)^2 \right\} \quad (3.11)$$

3.2.2 The Performance of a Multiprocessor System with Special Purpose Processors

This system is similar to the one described in Figure 3.5. Let us denote by C_i the capacity of each processor in the i th subsystem. Let m_i and $W_n^{(i)}$ be the number of processors and the amount of work remaining

at the n th epoch in the i th subsystem, respectively. By applying the same arguments in 3.2.1, we express $W_{n+1}^{(i)}$ as follows:

$$W_{n+1}^{(i)} = \begin{cases} W_n^{(i)} + S_i - C_i m_i A & \text{if } W_n^{(i)} + S_i - C_i m_i A > 0 \\ 0 & \text{if } W_n^{(i)} + S_i - C_i m_i A < 0 \end{cases}$$

Under heavy traffice conditions, we derive the expression for the waiting time in the queue in the case of a single processor system with interarrival time of tasks being $C_i m_i A$ and the execution time for a job being S . Since S and $C_i m_i A$ are exponentially distributed, then:

$$E[m_i C_i A] = \frac{C_i m_i}{\lambda}$$

$$E[S_i] = \frac{1}{\alpha_i}$$

$$E[S_i^2] = \frac{2}{\alpha_i^2}$$

Consequently, we get that the mean waiting time in queue for a single processor system is equivalent to the average amount of work remaining in the subsystem $\bar{W}_S^{(i)}$. Moreover, $\bar{W}_S^{(i)}$ is given by:

$$\bar{W}_S^{(i)} = \frac{(\lambda/C_i m_i) E(S_i^2)}{2} \left(\frac{1}{1-\rho_i} \right)$$

$$= \frac{\left(\frac{\lambda}{C_i m_i} \right) \left(\frac{2}{\alpha_i^2} \right)}{2} \left(\frac{1}{1-\rho_i} \right)$$

where

$$\rho_i = (\lambda/C_i m_i) \left(\frac{1}{\alpha_i}\right)$$

Thus the total average amount of work remaining, \bar{W}_S , is:

$$\begin{aligned} \bar{W}_S &= \sum_{i=0}^N \bar{W}_S^{(i)} = \sum_{i=0}^N \frac{\lambda}{C_i m_i \alpha_i^2} \left(\frac{1}{1 - \frac{\lambda}{\alpha_i C_i m_i}} \right) \\ &= \sum_{i=0}^N \left(\frac{1}{\alpha_i} \right) \left(\frac{\rho_i}{1 - \rho_i} \right) \end{aligned} \quad (3.12)$$

or

$$\bar{W}_S = \lambda \sum_{i=0}^N \left(\frac{1}{\alpha_i} \right) \left(\frac{1}{\alpha_i C_i m_i - \lambda} \right) \quad (3.13)$$

We can minimize the value of \bar{W}_S with respect to m_i under the condition $m = \sum_{i=0}^N m_i$. Since $\bar{W}_S(m_0, m_1, \dots, m_N)$ is a convex function, we can use the Lagrange method. Let

$$\bar{W}_S(m_0, m_1, \dots, m_N, \beta) = \bar{W}_S + \beta \left(\sum_{i=0}^N m_i - m \right).$$

To solve $\frac{\partial \bar{W}_S}{\partial m_i} = 0$, we get

$$\frac{\lambda C_i}{(\alpha_i C_i m_i - \lambda)^2} = \beta$$

Therefore,

$$m_i = \frac{1}{\alpha_i C_i} \left\{ \lambda + \sqrt{\frac{\lambda C_i}{\beta}} \right\}. \quad (3.14)$$

Since $m = \sum_{i=0}^N m_i$,

$$m = \lambda \sum_{i=0}^N \frac{1}{\alpha_i C_i} + \sum_{i=0}^N \sqrt{\frac{\lambda}{\beta \alpha_i^2 C_i}} = \lambda \sum_{i=0}^N \frac{1}{\alpha_i C_i} + \sqrt{\frac{1}{\beta}} \sum_{i=0}^N \sqrt{\frac{\lambda}{\alpha_i^2 C_i}}$$

Thus

$$\sqrt{\frac{1}{\beta}} = \frac{m - \lambda \sum_{i=0}^N \frac{1}{\alpha_i C_i}}{\sum_{i=0}^N \sqrt{\frac{\lambda}{\alpha_i^2 C_i}}} \quad (3.15)$$

Therefore, from eqs. (3.14) and (3.15), we get:

$$m_i = \frac{1}{\alpha_i C_i} \left\{ \lambda + \sqrt{C_i} \left(\frac{m - \lambda \sum_{j=0}^N \frac{1}{\alpha_j C_j}}{\sum_{j=0}^N \sqrt{\frac{1}{\alpha_j^2 C_j}}} \right) \right\} \quad (3.16)$$

Furthermore, the minimum value of \bar{W}_S is:

$$\text{Min}(\bar{W}_S) = \lambda \sum_{i=0}^N \left[\lambda + \sqrt{C_i} \left(\frac{m - \lambda \sum_{j=0}^N \frac{1}{\alpha_j C_j}}{\sum_{j=0}^N \sqrt{\frac{1}{\alpha_j^2 C_j}}} \right) - \lambda \right]^{-1} \left(\frac{1}{\alpha_i} \right)$$

$$= \lambda \sum_{i=0}^N \frac{\sum_{j=0}^N \sqrt{\frac{1}{\alpha_j^2 C_j}}}{m \sqrt{C_i} - \lambda \sqrt{C_i} \sum_{j=0}^N \left(\frac{1}{\alpha_j C_j} \right)} \left(\frac{1}{\alpha_i} \right)$$

$$= \lambda \left(\sum_{j=0}^N \sqrt{\frac{1}{\alpha_j^2 C_j}} \right) \cdot \sum_{i=0}^N \left\{ \left(\frac{1}{\sqrt{C_i}} \right) \left(\frac{1}{\alpha_i} \right) \left(\frac{1}{m - \lambda \sum_{j=0}^N \left(\frac{1}{\alpha_j C_j} \right)} \right) \right\} .$$

Therefore,

$$\text{Min}(\bar{W}_S) = \left(\sum_{i=0}^N \sqrt{\frac{1}{\alpha_i^2 C_i}} \right)^2 \left(\frac{1}{\frac{m}{\lambda} - \sum_{j=0}^N \frac{1}{\alpha_j C_j}} \right) \quad (3.17)$$

If $C_i = C$ for all i , then eq. (3.16) can be simplified to:

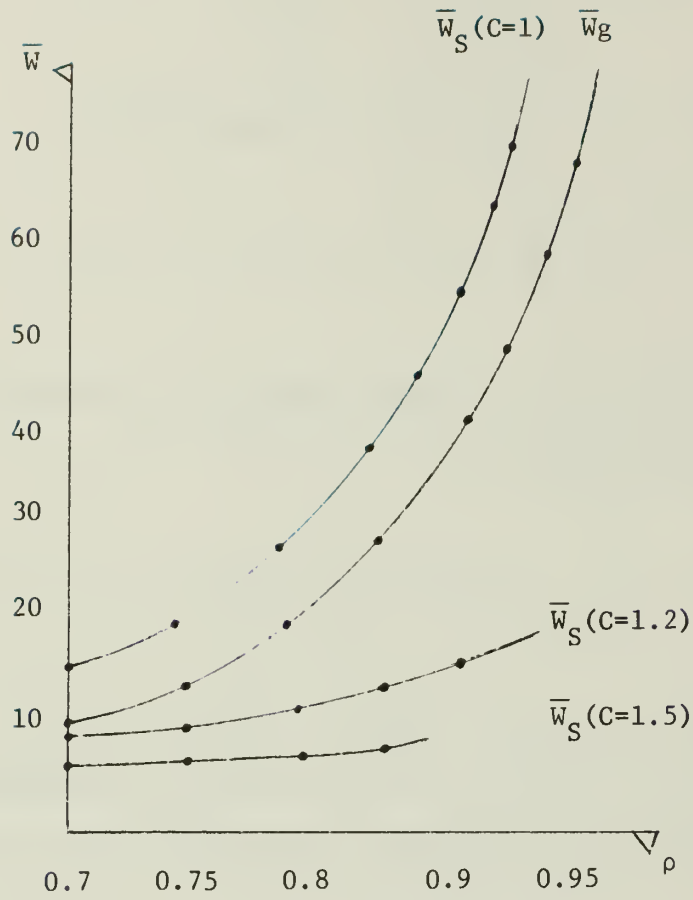
$$m_i = \frac{m}{\sum_{j=0}^N \frac{1}{\alpha_j}} \cdot \frac{1}{\alpha_i} \quad (3.18)$$

Obviously, when $C_i = C$, the value of m_i which minimize \bar{W}_S is independent of C . In this case, $\text{Min}(\bar{W}_S)$ is given by:

$$\text{Min}(\bar{W}_S) = \frac{1}{C} \cdot \left(\sum_{j=0}^N \frac{1}{\alpha_j} \right)^2 \left(\frac{m}{\lambda} - \frac{1}{C} \sum_{j=0}^N \frac{1}{\alpha_j} \right)^{-1} \quad (3.19)$$

Figure 3.7 shows the curves $\bar{W}_S = \text{Min}(\bar{W}_S)$, and \bar{W}_g in the case of $N + 1 = 3$, $\left(\frac{1}{\alpha_i}\right) = i + 1$ for $i = 0, 1, 2$, $C_i = C$ and $m = 10$. As expected, under heavily loaded conditions, the optimized system with special purpose processors behaves better. Even in the case $C = 1$ (i.e., no improvement of processor speed), the value of \bar{W}_S is almost comparable to that of \bar{W}_g . Obviously, we obtain this result since the architectural flexibility in scheduling does not make much difference under heavily loaded conditions.

Figure 3.8 describes another case where $N + 1 = m = 3$, $\left(\frac{1}{\alpha_i}\right) = (0.9)^i$, $C_i = C$ for $i = 0, 1, 2$. Therefore, $m_i = 1$ for all i . Since all subsystems consist of a single processor, eq. (3.12) holds for any value of ρ_i (not necessarily $\rho_i \approx 1$ for all i). Obviously \bar{W}_S cannot be optimized in this case.



where

$$\text{for } \bar{W}_g: \quad \rho = \frac{\lambda}{5}, \quad \bar{W}_g = \frac{25}{6} \left(\frac{1}{\frac{1}{\rho} - 1} \right), \quad m_1 = 5, \quad m_2 = 10, \quad m_3 = 15$$

$$\text{for } \bar{W}_S: \quad \rho_i = \frac{\lambda}{5C}, \quad \bar{W}_S = 6 \left(\frac{1}{\frac{C}{\rho} - 1} \right)$$

Figure 3.7 \bar{W}_g and \bar{W}_S under Heavily Loaded Condition

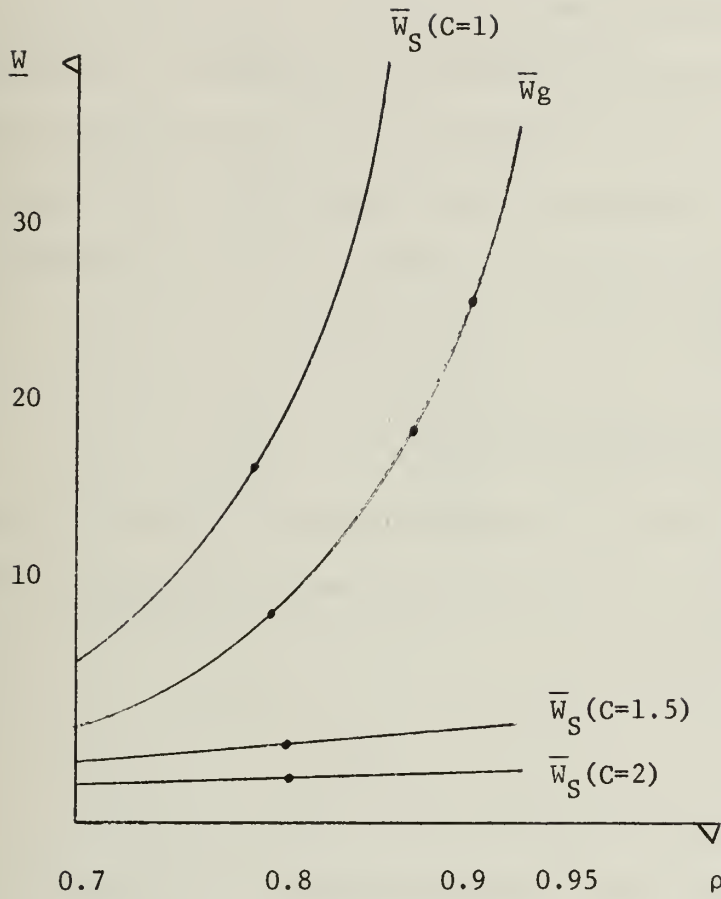


Figure 3.8 \bar{W}_g and \bar{W}_S under Heavily Loaded Condition

4. THE SYSTEM WITH SERIAL INPUT PROCESS

In this section, we consider the model in which the interarrival time of jobs is exponentially distributed with parameter λ . Each of the arrived jobs consists of $N + 1$ different types of tasks which must be executed in sequence. The amount of work for a type i task is exponentially distributed with parameter α_i . That is, if S_i is the amount of work for a type i task, $\Pr\{S_i \leq t\} = 1 - e^{-\alpha_i t}$, $t \geq 0$.

4.1 The Performance of a Multiprocessor System with General Purpose Processors

This system consists of m general purpose processors. It can be described as in Figure 4.1. Since the tasks must be executed in order, the execution time of a job S is:

$$S = S_0 + S_1 + \dots + S_N$$

where S_i is the amount of work for a type i task.

We again assume heavily loaded condition as in section 3.2. Since the model is the same as that in section 3.2 in this case, from eq. (3.11), the average amount of work in the system is

$$\bar{W}_g = \frac{1}{2(1-\rho)} \left\{ \rho \sum_{i=0}^N \frac{1}{\alpha_i} + (\lambda/m) \sum_{i=0}^N \left(\frac{1}{\alpha_i} \right)^2 \right\}$$

where $\rho = (\lambda/m) \sum_{i=0}^N \frac{1}{\alpha_i}$.

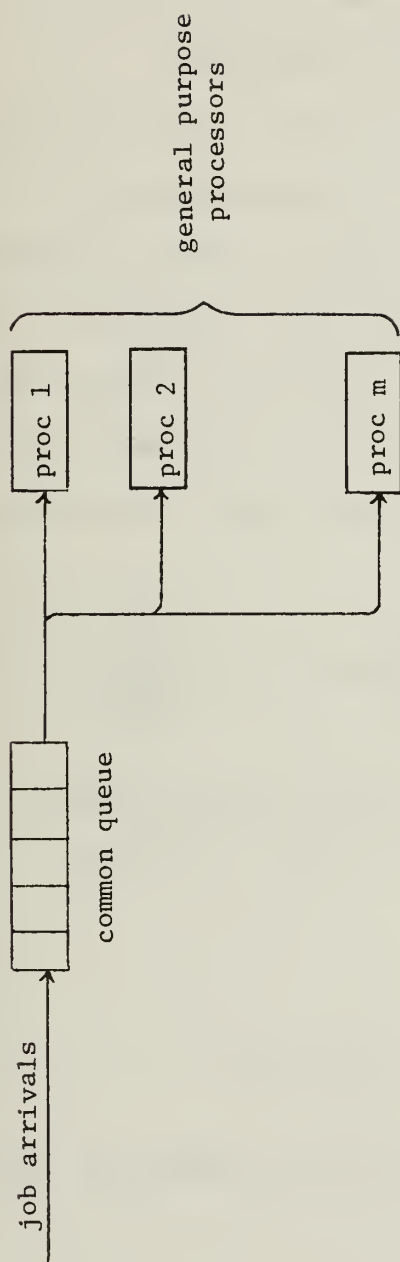


Figure 4.1 System with General Purpose Processors

4.2 The Performance of a Multiprocessor System with Special Purpose Processors

This system is described as in Figure 4.2. The i th subsystem consists of m_i special purpose processors where $m = m_0 + m_1 + \dots + m_N$. Since a special purpose processor can execute only one type of tasks, the job must go through the series of $N + 1$ subsystems.

Let us consider the number of tasks in the i th subsystem with C_i being the capacity of a processor in the i th subsystem ($C_i \geq 1$). The execution time of a job at the processor in the i th subsystem is exponentially distributed with parameter $C_i \alpha_i$. Therefore, by Burke's theorem [5], the interarrival time of jobs to the i th subsystem is also exponentially distributed with parameter λ . If we assume heavily loaded condition (all processors are busy all the time), the average amount of work in the i th subsystem $\bar{W}_S^{(i) '}$ is:

$$\bar{W}_S^{(i) '} = \frac{\lambda}{C_i m_i \alpha_i^2} \left(\frac{1}{1 - \rho_i} \right) \quad \text{where} \quad \rho_i = \frac{\lambda}{C_i m_i \alpha_i}$$

However, the jobs in the i th subsystem have additional amount of work. (That is, tasks for $i + 1, i + 2, \dots, N$ types.) Therefore, the amount of work in the i th subsystem $W_S^{(i)}$ is

$$W_S^{(i)} = \bar{n}_i \sum_{j=i+1}^N \frac{1}{\alpha_j} + W_S^{(i) '}$$

where \bar{n}_i is the mean number of jobs in the i th subsystem. Since

$\bar{n}_i = \frac{\rho_i}{1 - \rho_i}$ [3], then we get:

$$W_S^{(i)} = \frac{\rho_i}{1 - \rho_i} \sum_{j=i+1}^N \frac{1}{\alpha_j} + \frac{\lambda}{C_i m_i \alpha_i^2} \left(\frac{1}{1 - \rho_i} \right) .$$

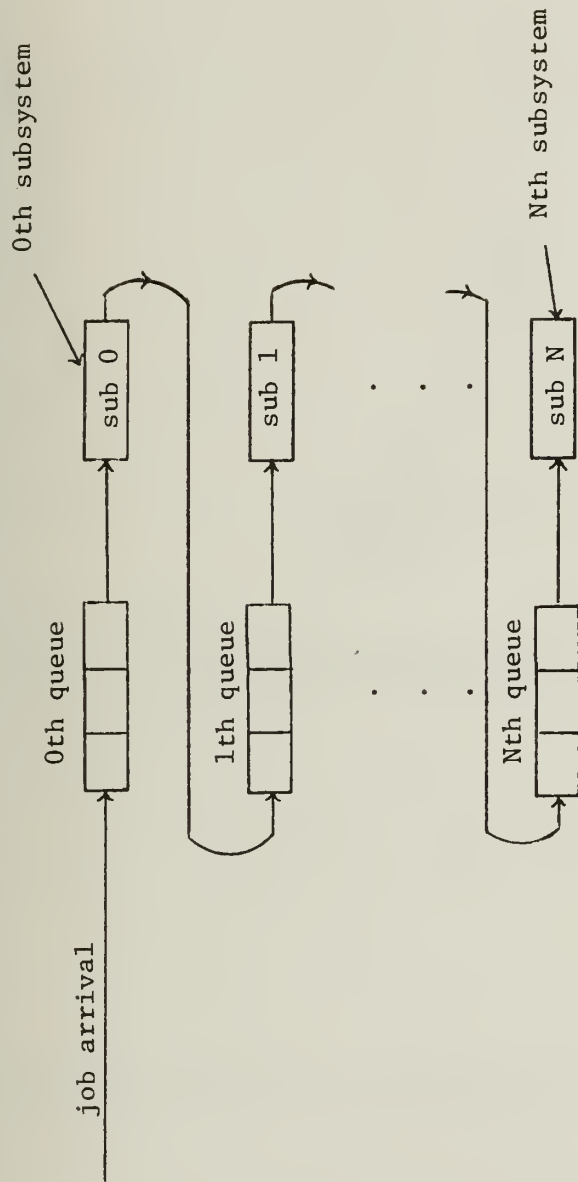


Figure 4.2 Multiprocessor System with Special Purpose Processors

Therefore, the total average amount of work \bar{W}_S is:

$$\begin{aligned}\bar{W}_S &= \sum_{i=0}^N W_S^{(i)} = \sum_{i=0}^N \left(\frac{\rho_i}{1-\rho_i} \right) \left(\sum_{j=i+1}^N \frac{1}{\alpha_j} \right) + \frac{1}{\alpha_i} \left(\frac{\rho_i}{1-\rho_i} \right) \\ &= \sum_{i=0}^N \sum_{j=1}^N \frac{1}{\alpha_j} \left(\frac{\rho_i}{1-\rho_i} \right) .\end{aligned}\tag{4.1}$$

where

$$\rho_i = \frac{\lambda}{C_i m_i \alpha_i}$$

If $C_i = C$ for all i , then

$$\bar{W}_S = \sum_{i=0}^N \sum_{j=1}^N \frac{1}{\alpha_j} \left(\frac{\frac{\lambda}{m_i \alpha_i}}{C - \frac{\lambda}{m_i \alpha_i}} \right) .\tag{4.2}$$

Figure 4.3 describes the behaviors \bar{W}_g and \bar{W}_S in the case of $N + 1 = m = 3$, $\left(\frac{1}{\alpha_i} \right) = (0.9)^i$, $C_i = C$.

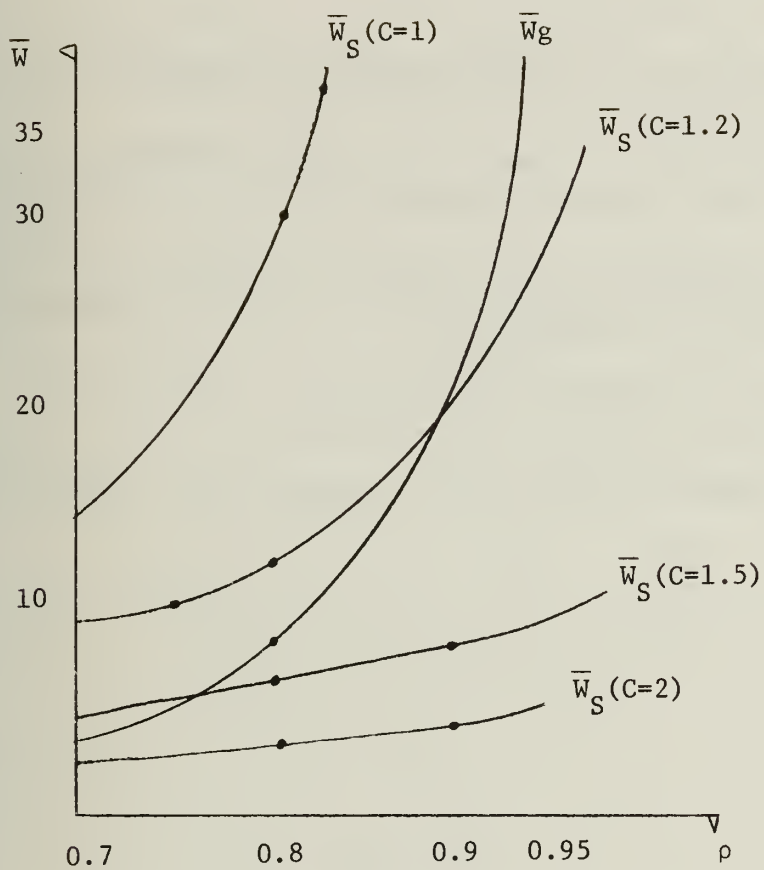


Figure 4.3 \bar{W}_g and \bar{W}_S under Heavily Loaded Condition

LIST OF REFERENCES

- [1] Philip, H. and Enslow, J. R., "Multiprocessor Organization - A Survey," Computing Surveys, Vol. 9, No. 1, March 1977.
- [2] Wulf, W. A. and Bell, G. C., "Cmmp - A Multi-Mini-Processor," Proc. of AFIPS 1972 Fall Joint Computer Conference, Vol. 41, 1972, pp. 765-777.
- [3] Baskin, H. B., Borgenon, B. R. and Roberts, R., "Prime - A Modular Architecture for Terminal-Oriented Systems," AFIPS 1972 Spring Joint Computer Conference, 1972, pp. 431-437.
- [4] Coffman and Denning, "Operating Systems Theory," Prentice-Hall, Englewood Cliffs, New Jersey, pp. 157-162.
- [5] Kleinrock, L., "Queueing Systems," Vol. 1, Wiley, 1975, pp. 147-160.

APPENDIX

In this appendix, we derive the expressions given by eq. (3.5), the average amount of work in equilibrium state for the system consisting of m general purpose processors (Figure 3.2). Therefore, we need to solve for $\lim_{Z \rightarrow 1} \frac{dP(Z)}{dZ}$, which is given:

$$\lim_{Z \rightarrow 1} \frac{dP(Z)}{dZ} = \lim_{Z \rightarrow 1} \left\{ \frac{A(Z)F(Z) \cdot (A(Z) - Z^m) - A(Z)F(Z)(A'(Z) - mZ^{m-1})}{(A(Z) - Z^m)^2} \right\}.$$

Let $N(Z)$ and $D(Z)$ be the numerator and denominator respectively in the fraction above. Since $\lim_{Z \rightarrow 1} N(Z) = F'(1) - F'(1) = 0$ and $\lim_{Z \rightarrow 1} D(Z) = 0$, we need to find the expression for both $\frac{dN(Z)}{dZ}$ and $\frac{dD(Z)}{dZ}$. Clearly,

$$\frac{dD(Z)}{dZ} = 2(A'(Z) - mZ^{m-1})(A(Z) - Z^m).$$

Therefore,

$$\lim_{Z \rightarrow 1} \frac{dD(Z)}{dZ} = 0.$$

Also,

$$\begin{aligned} \frac{dN(Z)}{dZ} &= 2AA'F' + A^2F'' - mZ^{m-1}(A'F + AF') - Z^m(A''F + 2A'F' + AF'') \\ &\quad + FA(m(m-1)Z^{m-2}) + mZ^{m-1}(AF' + A'F) \\ &= F''(A^2 - Z^m A) + F'(2AA' - 2A'Z^m) + F(m(m-1)Z^{m-2}A - Z^m A') \quad (A-1) \end{aligned}$$

Since $\lim_{Z \rightarrow 1} \frac{dN(Z)}{dZ} = \lim_{Z \rightarrow 1} \{F'(2A' - 2A')\} = 0$, we need to find $\frac{d^2N(Z)}{dZ^2}$ and $\frac{d^2D(Z)}{dZ^2}$

Differentiating the expression in eq. (A-1), we have

$$\begin{aligned} \frac{d^2 N(Z)}{dZ} &= F''' (A^2 - Z^m A) \\ &+ F'' (2AA' - Z^m A' - mZ^{m-1} A + 2AA' - 2A' Z^m) \\ &+ F' (2A'^2 + 2AA'' - 2(Z^m A'' + mZ^{m-1} A') + m(m-1)Z^{m-2} A - Z^m A'') \\ &+ F(m(m-1)Z^{m-2} A' + m(m-1)(m-2)Z^{m-3} A - Z^m A''' - mZ^{m-1} A'') \quad . \end{aligned}$$

Since $F(1) = 0$, $A(1) = 1$,

$$\begin{aligned} \lim_{Z \rightarrow 1} \frac{d^2 N(Z)}{dZ} &= F''(1) (4A'(1) - m - A'(1) - 2A'(1)) \\ &+ F'(1) \cdot (2A'^2(1) + 2A''(1) - 2A''(1) - 2mA'(1) + m(m-1) - A''(1)) \\ \lim_{Z \rightarrow 1} \frac{d^2 D(Z)}{dZ} &= 2 \lim_{Z \rightarrow 1} (A(Z) - mZ^{m-1})^2 = 2 (A'(1) - m)^2 = 2F'(1) \quad . \end{aligned}$$

Hence,

$$\begin{aligned} N''(1) &= F''(1) \cdot F'(1) + F'(1) (2A'(1)F'(1) + m(m-1) - A''(1)) \quad , \\ D''(1) &= 2F'^2(1) \quad . \end{aligned}$$

Thus,

$$\lim_{Z \rightarrow 1} \frac{dP(Z)}{dZ} = \frac{F''(1)}{2F'(1)} + \frac{2A'(1)F'(1) + m(m-1) - A''(1)}{2F'(1)} \quad (A.2)$$

In other words, the mean number of tasks in the system is:

$$\bar{n} = \frac{F''(1)}{2F'(1)} + \frac{m(m-1)}{2F'(1)} + \frac{-A''(1)}{2F'(1)} + A'(1) \quad (A.3)$$

where $F(Z) = \sum_{i=0}^{m-1} \pi_i (Z^i - Z^m)$. Particularly, if $m = 1$, then

$F'(1) = A'(1) - 1 = -(1-\rho)$, where ρ is the mean number of arrivals during

one time slot, and $F''(1) = 0$. Hence,

$$\bar{n} = \rho + \frac{A''(1)}{2(1-\rho)}.$$

We now solve the value of $\frac{F''(1)}{F'(1)}$. According to eq. (3.3), $P(Z)$ is expressed as follows:

$$P(Z) = \frac{A(Z)F(Z)}{A(Z)-Z^m}$$

Since $P(Z)$ is analytic and bound within the unit circle, all zeros of the denominator within the unit circle must coincide with those of the numerator.

While by Roache's Theorem, $F(Z)$ can be expressed as follows:

$$F(Z) = K(Z-1)(Z-Z_0)\dots(Z-Z_{m-2})$$

where Z_i is the root of $A(Z) = Z^m$ inside of unit circle. Thus,

$$F'(Z) = K \left\{ \prod_{i=0}^{m-2} (Z-Z_i) + (Z-1) \sum_{i=0}^{m-2} \prod_{j \neq i} (Z-Z_j) \right\},$$

$$F'(1) = K \prod_{i=0}^{m-2} (1-Z_i).$$

Therefore,

$$F''(Z) = K \left\{ \sum_{i=0}^{m-2} \prod_{j \neq i} (Z-Z_j) + \sum_{i=0}^{m-2} \prod_{j \neq i} (Z-Z_j) + (Z-1) \left(\sum_{i=0}^{m-2} \prod_{j \neq i} (Z-Z_j) \right)' \right\},$$

$$F''(1) = K \left\{ 2 \sum_{i=0}^{m-2} \prod_{j \neq i} (1-Z_j) \right\}.$$

Then, we get:

$$\begin{aligned} \frac{F''(1)}{F'(1)} &= \frac{2K \sum_{i=0}^{m-2} \prod_{j \neq i} (1-Z_j)}{K \prod_{i=0}^{m-2} (1-Z_i)} \\ &= 2 \sum_{i=0}^{m-2} \frac{1}{(1-Z_i)} \end{aligned}$$

Therefore, the mean number of tasks is:

$$\bar{n} = \sum_{i=0}^{m-2} \frac{1}{(1-Z_i)} - \frac{m(m-1)}{2(m-A'(1))} + A'(1) + \frac{A''(1)}{2(m-A'(1))}$$

where Z_0, Z_1, \dots, Z_{m-2} are the zeros of $(1 - \frac{Z^m}{A(Z)})$, in the unit circle.

BIBLIOGRAPHIC DATA Title and Subtitle	1. Report No. UIUCDCS-R-78-891	2.	3. Recipient's Accession No.
	PERFORMANCE EVALUATION OF MULTIPROCESSOR SYSTEMS CONTAINING SPECIAL PURPOSE PROCESSORS		5. Report Date January 1978
6.			
Author(s) Haruaki Yamazaki	8. Performing Organization Rept. No.		
Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois 61801		10. Project/Task/Work Unit No.	
		11. Contract/Grant No. MCS 73-07980 A03	
Sponsoring Organization Name and Address National Science Foundation Washington, D.C.		13. Type of Report & Period Covered	
		14.	

Supplementary Notes

ABSTRACT

In this paper, multiprocessor systems are characterized by the types of processors in the system. Some multiprocessor systems consist of identical general purpose processors which share the input job load under the control of the job scheduler. Other multiprocessor systems consist of special purpose processors each of which is designed to execute a particular type of job. Both types of multiprocessor systems are modeled queueing theoretically, and their performance is evaluated to determine their relative architectural merits. In the case of the multiprocessor system with special purpose processors, the optimal architecture is discussed.

Key Words and Document Analysis. 17a. Descriptors

multiprocessor systems, functionally dedicated modules.

Identifiers, Open-Ended Terms

OSAI Field/Group

Availability Statement	19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages
	20. Security Class (This Page) UNCLASSIFIED	22. Price

OCT 2 1980



UNIVERSITY OF ILLINOIS-URBANA
S10.84 IL6R no. C002 no.886-893(1977
Generating binary trees lexicographically



3 0112 088403594